REMARKS

Claims 5-6 and 8-10 have not been rejected and are believed to stand allowable if rewritten in independent form including all of the limitations of the base claim. Claims 1-4, and 7 stand rejected under 35 USC §103(a) as being unpatentable over Rojas et al. (Rojas), U.S. patent application US 2003/0223416 A1 in view of Crocker et al. (Crocker) U.S. patent 5,953,746. Claims 11-19 stand rejected under 35 USC §103(a) as being unpatentable over Rojas et al. (Rojas), U.S. patent application US 2003/0223416 A1 in view of Crocker et al. (Crocker) U.S. patent 5,953,746 and further view of Craddock et al., (Craddock) U.S. patent application US 2003/0046505 A1

Claims 1, 5, 6, 11, 16, 17, and 18 have been amended to more clearly state the invention. Claim 8 has been cancelled. Claim 5 has been rewritten in independent form including all of the limitations of the base claim. Independent claim 1 has been amended to include the subject matter of dependent claim 8. Independent claims 11, and 17 have been amended to include the subject matter limitations of dependent claim 8.

Each of the pending claims 1-7 and 9-19, as amended, is believed to be in condition for allowance and allowance is respectfully requested.

Reconsideration and allowance of each of the pending claims 1-7, and 9-19, as amended, is respectfully requested.

Rojas et al. (Rojas), U.S. patent application US 2003/0223416 A1 discloses a switch for use with an InfiniBand network. The switch includes a crossbar

that redirects packet-based data based on a forwarding table. At least one port that

receives data from a network and selectively transfers that data to the crossbar using a

variable number of virtual lanes. A state machine controls the changing of the number

of virtual lanes. Rojas discloses:

> [0044] A state machine 512 dynamically reallocates VL buffer space in the line 406 by managing memory in the input buffer 508, which as noted above physically contains the virtual lanes 306 (see FIG. 3). The state machine 512 is responsive to flow control messages (stripped and forwarded by the PLI 502) for dividing the available memory in the input buffer 508 among the virtual lanes 306 (see FIG. 3). In general, a link 406 supports 1, 2, 4, or 8 virtual lanes 306. For each link 406, the virtual lanes are preferably implemented in a single, contiguous memory.
>
> [0045] The amount of memory allocated to each virtual lane is initially determined by dividing the total size of the memory by the number of VL configured (as defined by the OperationalVL attribute). As the switch 500 receives packets they are stored (partially or entirely) in the input buffer 508, in the space defined for the packet's virtual lane. In an active link 406, several packets can reside in the virtual lanes in the input buffer 508. An SMP can, at any time, change the number of configured virtual lanes forcing a redistribution of the amount of memory allocated to each virtual lane. The state machine 512 controls this process.
>
> [0052] In step 622, a virtual lane is checked to determine whether the data in an identified virtual lane is less than the new required size. If the data is less than the new size, the method proceeds to step 624 and a check is made to determine whether the virtual lane is idle. Should the identified lane not be idle, the method returns to step 618. Assuming that the lane is not active, it is resized and marked complete (either in a register or in some data structure that described the virtual lane) in step 626. The method then proceeds to step 628, a next lane is selected and the method returns to step 618 to process the remaining virtual lanes to make room for the new virtual lanes to be added in step 630.

Crocker et al. (Crocker) U.S. patent 5,953,746 discloses a method and

system for dynamically sizing a dedicated memory in a shared memory buffer

architecture. At initial boot, system BIOS programs control register to allocate a

dedicated memory of a desired size. The size of the dedicated memory allocated is

dependent on the performance requirements. If after initial boot, the performance

requirements change, it may necessitate a change in dedicated memory size. By

reprogramming the control registers, the dedicated memory size is dynamically

changed without any manual manipulation of internal components. Crocker at column

4, lines 44-57 states:

A plurality of programmable bits shown in FIG. 3 as control register 22,
control register 23, and control register 24 store a control information which affects
arbitration and the claiming of addresses by the memory controller 20. Control register
22 is programmed to indicate whether a graphics add-in card is present on the 110 bus.
If an add-in card is present on the I/O bus, it will be necessary to disable graphics
controller 15 and the allocation of frame buffer 12 in favor of the add-in card. In one
embodiment, allocating an aperture size of zero will create no frame buffer in the
physical memory. With no allocated frame buffer, the graphics controller will not claim
addresses sent to the I/O bus. Control register 23 holds a value indicating the row base
address for frame buffer 12 in physical memory 13.

Craddock et al., (Craddock) U.S. patent application US 2003/0046505

discloses apparatus and method for swapping out real memory by inhibiting

input/output (I/O) operations to a memory region. The apparatus and method provide a

mechanism in which a quiesce indicator is provided in a field containing the current

outstanding I/O count associated with the memory region whose real memory is to be

swapped out. The current I/O field and the quiesce indicator are used as a means for

communicating between a shared resource arbitrator and a guest consumer. When the

quiesce indicator is set, the guest consumer is informed that it should not send any

further I/O operations to that memory region. When the number of pending I/O

operations against the memory region is zero, a valid bit in a protection table is set to

invalid, and the real memory associated with the memory region may be swapped out.

Thereafter, when the memory region is swapped back in, an address translation table is

updated, the valid bit is reset, and the quiesce indicator is reset so that further I/O

operations to the memory region may occur. Craddock discloses:

[0124] There may be instances when the real memory backing a memory region needs to be temporarily swapped out, such as in an environment where operating system (OS) images (called guests) themselves are virtualized by a Hypervisor (i.e. guest real memory is the Hypervisor's virtual memory). That is, in order to share the processing capabilities of a system it is possible to have multiple Operating systems running on the same hardware. Each operating system thinks it owns the hardware resources and has no knowledge of the other operating systems that are running on this hardware. Each operating system has its own address space for accessing memory. In this environment it is necessary for a controlling entity to arbitrate for access to the shared hardware resources and to translate the "virtual addresses" that each operating system is aware of into the real addresses that are implemented in the hardware. This controlling entity is typically called a Hypervisor. The virtualization of the operating system allows an operating system to co-exist with other operating systems while using the same hardware.

[0127] To enable real memory swap support, the guest channel interface (CI) provides an interface to allow the consumer within the guest to communicate directly with the supporting Hypervisor. This interface may then be used by the guest consumer (not the channel interface (CI), as only the consumer has the appropriate knowledge) to guarantee that there will be no accesses to a memory region while it is swapped out. The consumer may use the interface while the channel interface cannot because the consumer is responsible for the upper-level protocol running above the SAN protocol. The CI is only aware of the SAN protocol layers. The consumer can guarantee that there will be no accesses to the memory region by not initiating any operations that access the memory region. The consumer communicates with the Hypervisor to inform the Hypervisor when it is safe to swap-out the memory region and also when the memory region needs to be paged back in so that operations may resume.

Reconsideration and allowance of each of the pending claims 1-7 and 9-19, as amended, is respectfully requested.

The invention enable VL buffer resource resident on an adapter to be dynamically allocated between either ports, or virtual lanes within a port, while operating. An exemplary set of metrics is provided that allow applications to determine when buffer changes may be advantageous. These techniques contribute to the following advantages: An adapter designer can choose to implement a smaller on-chip

set of VL Buffer knowing that it can be adjusted in the future as needed; reduces overall hardware cost; provides the mechanisms to respond to network congestion problems; and can adjust buffering to match desired Quality of Service policies.

Independent claim 1, as amended, recites a method for implementing dynamic Virtual Lane (VL) buffer reconfiguration in a channel adapter. Independent claim 1, as amended, recites the steps of providing a hypervisor for monitoring buffer resources, and using said hypervisor for writing change requests to respective ones of each said second register and said third registers. Thus, independent claim 1, as amended, further defines the invention to include the subject matter of dependent claim 8, which has not been rejected.

Independent claim 1, as amended, further recites the steps including providing at least one second register for communicating a current port buffer size; one said second register associated with each physical port of the channel adapter; and utilizing said second register for receiving change requests for adjusting said current port buffer size for an associated physical port. None of the cited references suggest dynamic adjusting of the port buffer size for an associated physical port as taught and claimed by applicants. These steps are not shown or suggested by the references of record including Rojos, Crocker, and Craddock. These limitations as recited in independent claim 1, as amended, are not disclosed or suggested by the combined teachings of Rojos, Crocker, and Craddock.

Thus, independent claim 1, as amended, is patentable.

Independent claims 11, and 17 respectively recite apparatus for

implementing dynamic Virtual Lane (VL) buffer reconfiguration and a computer program product for implementing dynamic Virtual Lane (VL) buffer reconfiguration in a channel adapter of a system area network. Each of the independent claims 11, and 17, as amended, is patentable for the same reasons as independent claim 1.

As amended, independent claim 11 defines registers of the invention and further recites the hypervisor for monitoring buffer resources; said hypervisor for writing change requests to said second register for adjusting said current port buffer size for an associated physical port; said hypervisor for writing change requests to said third register for adjusting said current VL buffer size for an associated VL; and channel adapter hardware for managing allocation of buffer space responsive to said change requests written by said hypervisor. These limitations as recited in independent claim 11, as amended, are not disclosed or suggested by the combined teachings of Rojos, Crocker, and Craddock.

Thus, independent claim 11, as amended, is patentable.

As amended, independent claim 17 recites the steps of: communicating an adapter buffer size and allocation capability for the channel adapter using a first register; communicating a current port buffer size using a second register; one said second register associated with each physical port of the channel adapter; communicating a current VL buffer size using a third register; one said third register associated with each VL of each said physical port of the channel adapter; monitoring buffer resources; writing a change request to one said second register for adjusting said current port buffer size for said associated physical port; and writing a change request

-13-

to one said third register for adjusting said current VL buffer size for said associated VL.

Only applicants teach the computer program product as recited in independent claim 17, as amended. More specifically only applicant teach monitoring buffer resources and writing a change request to one said second register for adjusting said current port buffer size for said associated physical port. These limitations as recited in independent claim 17, as amended, are not disclosed or suggested by the combined teachings of Rojos, Crocker, and Craddock.

Thus, independent claim 17, as amended, is patentable.

Dependent claims 2-4, 6-7, 9-10, 12-16 and 18-19 respectively depend from patentable claims 1, 11, and 17, further defining the invention. Each of the dependent claims 2-4, 6-7, 9-10, 12-16 and 18-19, as amended, is likewise patentable.
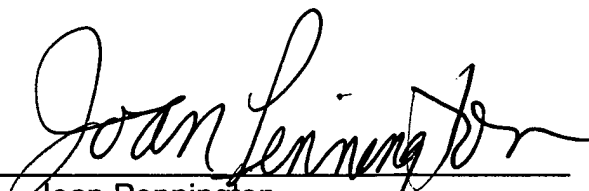
Applicants have reviewed all the art of record, and respectfully submit that the claimed invention is patentable over all the art of record, including the references not relied upon by the Examiner for the rejection of the pending claims.

It is believed that the present application is now in condition for allowance and allowance of each of the pending claims 1-7, and 9-19, as amended is respectfully requested. Prompt and favorable reconsideration is respectfully requested.

If the Examiner upon considering this amendment should find that a telephone interview would be helpful in expediting allowance of the present application, the Examiner is respectfully urged to call the applicants' attorney at the number listed below.

Respectfully submitted,

-14-

By: _____
Joan Pennington
Reg. No. 30,885
Telephone: (312) 670-0736